gtialign

November 4, 2014

Abstract

Align asynchronous GTI files to frame readout boundaries

1 Instruments/Modes

Instrument	Mode
EPIC	All
RGS	All
OM	Fast

Note that the primary use of this task is foreseen to be with the EPIC imaging mode and with the RGS. It will not be used by default for processing EPIC timing mode data, and its applicability to the OM fast mode is TBD.

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

The purpose of this task is to take a time series and a GTI file, and align the GTIs with the time bin edges.

As initially conceived, the task was restricted in applicability: the only time series input it would accept was the EXPOSU extensions of an EPIC event list; the input GTI series was aligned separately with each of these EXPOSU time series, the respective results being then written to separate extensions of the output file. This behaviour has been retained, but new parameters were added in version 2.9 to allow the task to be applied to generic OGIP-compliant time series and GTI files.

The problem which **gtialign** is intended to solve is illustrated in the figure below.

```
|        |              |        |           |             |             |
|   1    |      2       |   3    |     4     |      5      |      6      |
|        |              |        |           |             |             |


                  -----------------------------------
                 |                                   |
                 |                                   |
-----------------|                                   |---------------------

time -->
```

The top line represents the time boundaries for a given time series. In the case of the EXPOSU extensions the time bins are actually the CCD readout frames: in this case each vertical line represents the end of one frame readout cycle and the beginning of another. The bottom line represents a GTI, as produced for example from housekeeping data. ‘Up’ corresponds to a good time interval. Note that the time-series bins are not necessarily of constant duration.

Looking at time bin 2, we see that the GTI is valid only for part of this interval. To avoid uncertainty about the validity of bin 2, all of it should be excluded. Similarly, all events from bin 5 should be excluded as well. Thus the GTI should be adjusted as follows:

```
|        |              |        |           |             |             |
|   1    |      2       |   3    |     4     |      5      |      6      |
|        |              |        |           |             |             |


                         --------------------
                        |                    |
                        |                    |
------------------------|                    |-----------------------------

time -->
```

3.1 ‘Pipeline’ mode

This was the only mode of operation available before version 2.9. It is still the default.

In this mode, the **gtialign** task takes as input the GTI file to be aligned, and the event list to which the GTIs will eventually be applied. There should be exposure extension(s) in the event list, which will contain the frame readout times for the events in this file. This data will be used to adjust the GTI start and stop times as described above. The different CCDs in the instruments may have different frame integration start and stop times, so that the GTI file will have to be aligned separately for each CCD. Note that the frame integration time for a particular CCD can also vary within a given exposure.

Depending on the convention used (as recorded in the `TIMEPIXR` keyword of the `EVENTS` extension), the times associated with individual events can be set to the beginning, end or middle of the frame readout intervals (or, in principle, any value in between). This is also true of the exposure extension, which stores the frame readout times as an absolute time and an interval.

In the case of the event times, this task adjusts the GTIs such that any event in a frame with a good GTI will be accepted, no matter where in the frame readout interval this time is set.

However, for the exposure information, it is necessary to know what convention is being used for storing frame timing information. This will be determined by referring to the OGIP keyword `TIMEPIXR` in the header of the exposure extension. A value of 0 corresponds to time stamps referring to the beginning of the time interval, 1.0 to the end of the time interval, etc. If the keyword is not present, the default value is 0.5, i.e. the time stamp refers to the middle of the time interval.

The output of this task will be a FITSfile, with a GTI extension for each CCD in the event list. Each extension will have the name `STDGTI`*n*, where *n* is the CCD number for that extension. Where deemed useful, the extension will also contain CCD specific keywords such as `CCDID`, `CCDNODE`, and `FRMTIME`. The primary header of this file will contain all the standard XMM keywords, as copied from the original GTI file.

All GTI files will be OGIP compliant.

3.2 'Generic' mode

This is the new mode which was introduced in version 2.9. To access this mode you should set `style`='generic'. In this mode the task takes two inputs, supplied to parameters `ingtitable` and `tstable`, and generates a single GTI table as output, which is written to the dataset+table specified via parameter `outgtitable`. Otherwise the functioning is similar to the 'pipeline' mode.

An example command:

```
gtialign style=generic ingtitable=my_gtis.ds:STDGTI01 tstable=lccorr_out.ds:RATE
outgtitable=some_file.fits:NEW_GTI
```

4 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints

style	no	string	pipeline	pipeline—generic

Whether to access the pre- or post-2.9 mode. To access the pre-2.9 mode of operation, set `style` to 'pipeline'. In this mode, the inputs must be supplied via the parameters `gtitable` and `eventset`, and the output appears as one or more extensions named `STDGTI`*n* of the file pointed to by parameter `outset`. To access the new mode, set `style` to 'generic'. In this second mode, the inputs must be supplied via the parameters `ingtitable` and `tstable`, and the output is written to the binary table pointed to by parameter `outgtitable`.

gtitable	yes	table	none	valid table spec.

(This parameter is read if `style`='pipeline' (the default).) Specification of GTI extension to be aligned. This specification should follow the **dal** convention of `set:table`.

eventset	yes	dataset	none	valid file name.

(This parameter is read if `style`='pipeline' (the default).) File name of event list containing exposure extension(s).

outset	yes	dataset	none	valid file name.

(This parameter is read if `style`='pipeline' (the default).) File name of output file containing the aligned GTI extensions.

ingtitable	yes	table	none	valid table spec.

(This parameter is read if `style`='generic'.) Specification of GTI extension to be aligned. This specification should follow the **dal** convention of `set:table`.

tstable	yes	table	none	valid table spec.

(This parameter is read if `style`='generic'.) Specification of the input time series extension. This specification should follow the **dal** convention of `set:table`.

outgtitable	no	table	out_gti.ds:GTI	valid table spec.

(This parameter is read if `style`='generic'.) Specification of the output GTI extension. This specification should follow the **dal** convention of `set:table`.

5 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

badStyle *(error)*
The value of the `style` parameter was not recognized.

missingBlockName *(error)*
The user did not specify the extension name in the string supplied to parameter `outgtitable`.

OutsideFrames *(warning)*
Frame times do not span all GTIs.
corrective action: GTIs outside of frame times are not modified.

missingBlockName *(warning)*
The user did not specify the extension name in the string supplied to parameter `ingtitable` or `tstable`.
corrective action: The task takes the first block in the dataset.

6 Input Files

- If `style`='pipeline':
  1. OGIP-compliant GTI file, such as those produced by **hkselect**.
  2. FITS event list file, containing exposure extension(s), such as that produced by **evlistcomb**.

- If `style`=‘generic’:

  1. OGIP-compliant GTI file, such as those produced by **hkselect**.
  2. OGIP-compliant time series, such as those produced by **evselect**.

7 Output Files

- If style=‘pipeline’:

  1. FITS file containing OGIP-compliant GTI extensions, one for each CCD in the input event list.

- If style=‘generic’:

  1. FITS file containing a single OGIP-compliant GTI extension.

8 Algorithm

```
subroutine gtialign

   open gti_in
   open events_in
   open gti_out

   read gti_starts
   read gti_stops

   foreach ccd

      create gti_out extension

      read frame_times

      foreach frame_time
         frame_start = get_start(frame_time, time_del, time_offset)
         foreach gti_start
            if( gti_start < frame_start )
               gti_new_start = frame_start
               break gti_start loop
            endif
         next

         frame_end = get_end(frame_time, time_del, time_offset)
         foreach gti_stop
            if( gti_stop < frame_end )
               gti_new_stop = get_end(frame_time-1, time_del, time_offset)
               break gti_stop loop
            endif
         next
      next

      write gti_out extension, gti_new_start, gti_new_stop

   next

   close files

end subroutine  gtialign
```

9 Comments

None.

10 Future developments

This task could be enhanced with options such that it will align GTI files using exposure information which does not necessarily conform to the conventions described in this document.

References